Multimedia Signal Processing 1st Module and Fundamentals of Multimedia Signal Processing

date: January 16th, 2025

Ex.1 (Pt.12)

A signal $x(t)$ is sampled at 12kHz obtaining the signal $x[n]$, and we want to find the amplitude of its component at 4kHz but we cannot use the FFT.

Therefore we decide to emulate the behavior of the FFT filtering the signal with a cosinusoid $c[n]$ and a sinusoid $s[n]$ at the normalized frequency corresponding to the 4kHz components.

Both the $c[n]$ and $s[n]$ filters are realized as IIR filters activated from a single impulse.

1. [4 pts] Depict the pole zero plot for $c[n]$ and $s[n]$.

2. [4 pts] Define the difference equations for $c[n]$ and $s[n]$.

3. [4 pts] Describe how we shall apply the previously defined filters in order to measure the

intensity of the input signal at 4kHz detailed the length, in term of samples, of the input signal $x[n]$.

Ex.2 (Pt.9)

An audio signal is sampled at 50kHz and we want to change its sample rate to 40kHz.

1. [3 pts] Describe the procedure in order to decrease the signal sample rate detailing the downsampling and upsampling rates.
2. [3 pts] Depict in a plot the possible overlap between the replicas.
3. [3 pts] Provide a description of the ideal filter that you would use in order to filter the signal and describe a real filter that could approximate the ideal one.

CONTINUES ON THE BACK

Ex.3 (Pt.12) To be solved writing the MATLAB code on the sheet.

1) [2 pt] You are given the following plot of one period of a real DFT as a function of frequencies [Hz], starting from 0:

In particular, f1 = 1KHz, f2 = 2KHz, f3 = 8KHz. Define the signal x related to this DFT, considering a number of temporal samples equal to 10 times its period. The sampling rate of the system is 80KHz.

2) [4.5 pt] Design an FIR stop-band filter with the windowing method in order to remove the frequency f2 (but avoiding removing f1). Consider 89 filter samples and, as cut-off frequencies, the +20% and -20% of the frequency to remove.
   - Filter the signal x with this filter, defining the signal y.
   - What do you expect to find in the DFT of the signal y? What will happen in the DFT behaviour if we increase the number of filter samples?
   - Set the number of filter samples to 301 and repeat the filtering.

3) [3 pt] We want to increase the sampling rate from 80KHz to 120KHz. Implement the sampling rate conversion of the signal y(n). You choose all the parameters needed. Define the new obtained signal as w(n).

4) [2.5 pt] Plot the absolute value of the DFT of w(n) as a function of the normalized frequency axis (keep the initial Fs as normalization frequency).
   - Comment on the position of each peak you expect to find in W(f). Does the signal contain aliasing?

Solutions

Ex.1

In order to create a sinusoid or a cosinusoid at the normalized pulsation $\omega_s = 2\pi \frac{4kHz}{12kHz} = \frac{2}{3}\pi$ we need to place the two conjugate poles on the unit circle

The z-transform of the IIR filters will be:

$$C(z) = \sum_{n=0}^{\infty} \frac{1}{2}\left(e^{j\frac{2}{3}\pi n} + e^{-j\frac{2}{3}\pi n}\right) z^{-n} = \frac{1}{2}\sum_{n=0}^{\infty}\left(e^{j\frac{2}{3}\pi} z^{-1}\right)^{n} + \frac{1}{2}\sum_{n=0}^{\infty}\left(e^{-j\frac{2}{3}\pi} z^{-1}\right)^{n} =$$

$$= \frac{1}{2}\frac{1}{1-e^{j\frac{2}{3}\pi} z^{-1}} + \frac{1}{2}\frac{1}{1-e^{-j\frac{2}{3}\pi} z^{-1}} = \frac{1}{2}\frac{1-e^{-j\frac{2}{3}\pi} z^{-1} + 1 - e^{j\frac{2}{3}\pi} z^{-1}}{1-2\left(-\frac{1}{2}\right) z^{-1} + z^{-2}} = \frac{1-\cos\left(\frac{2}{3}\pi\right) z^{-1}}{1+z^{-1}+z^{-2}} = \frac{1+\frac{1}{2}z^{-1}}{1+z^{-1}+z^{-2}}$$

$$S(z) = \sum_{n=0}^{\infty} \frac{1}{2j}\left(e^{j\frac{2}{3}\pi n} - e^{-j\frac{2}{3}\pi n}\right) z^{-n} = \frac{1}{2j}\sum_{n=0}^{\infty}\left(e^{j\frac{2}{3}\pi} z^{-1}\right)^{n} - \frac{1}{2j}\sum_{n=0}^{\infty}\left(e^{-j\frac{2}{3}\pi} z^{-1}\right)^{n} =$$

$$= \frac{1}{2j}\frac{1}{1-e^{j\frac{2}{3}\pi} z^{-1}} - \frac{1}{2j}\frac{1}{1-e^{-j\frac{2}{3}\pi} z^{-1}} = \frac{1}{2j}\frac{1-e^{-j\frac{2}{3}\pi} z^{-1} - 1 + e^{j\frac{2}{3}\pi} z^{-1}}{1-2\left(-\frac{1}{2}\right) z^{-1} + z^{-2}} = \frac{\sin\left(\frac{2}{3}\pi\right) z^{-1}}{1+z^{-1}+z^{-2}} = \frac{\frac{\sqrt{3}}{2} z^{-1}}{1+z^{-1}+z^{-2}}$$

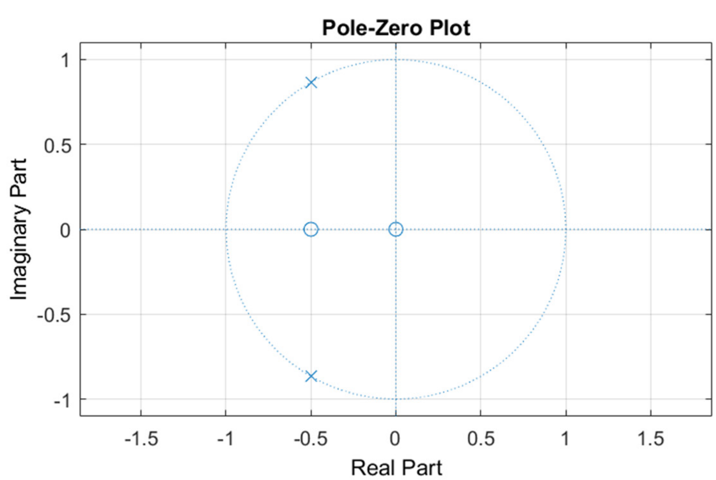


Figure 1 C(z) zeros-poles plot

$$c(n) = x(n) + \frac{1}{2}x(n-1) - c(n-1) - c(n-2)$$

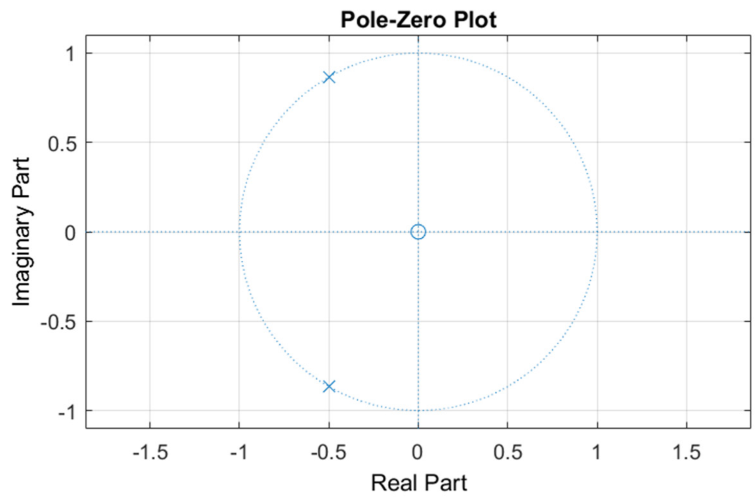


Figure 2 S(z) zeros-poles plot

$$s(n)=\frac{\sqrt{3}}{2}x(n-1)-s(n-1)-s(n-2)$$

To obtain the FFT result on N samples in imput for the selected pulsation $\omega=\frac{2}{3}\pi$, we need to compute the dot product between c(n) and s(n):

Real part: $\mathrm{Re}\left\{X\left(\omega=\frac{2}{3}\pi\right)\right\}=\sum_{n=0}^{N-1}x(n)\cdot c(n)$

Imaginary part: $\mathrm{Im}\left\{X\left(\omega=\frac{2}{3}\pi\right)\right\}=\sum_{n=0}^{N-1}x(n)\cdot s(n)$

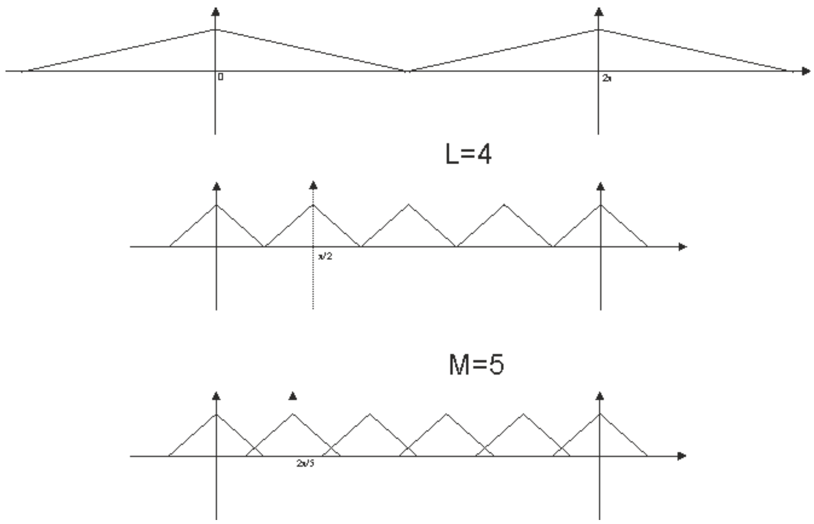
The amplitude will be as always the square root of the sum of the squared values of the two previous numbers.

Ex.2

Since $40kHz=50kHz\cdot\frac{4}{5}$ we need to apply an upsampler of order L=4 and a downsampler with order M=5.

To avoid aliasing with need to use a Low pass filter with a cutoff at $\omega_c=\frac{\pi}{5}$ for the signal upsampled at 200kHz, i.e. 20kHz.

Considering the upsample L=4 and a wrong low pass filter larger that $\omega_c = \frac{\pi}{5}$ and the following donwsampling we would get aliasing as depicted below:



Further details on course slides.

Ex.3

```
close all
clearvars

clc


%% 1. [2 pt]


% You are given the following plot of one period of a real DFT as a

% function of frequencies [Hz], starting from 0 (see exam)

% In particular, f1 = 1KHz, f2 = 2KHz, f3 = 8KHz.

% Define the signal x related to this DFT, considering a number of temporal samples

% equal to 10 times its period. The sampling rate of the system is 80KHz.


f1 = 1e3;

f2 = 2e3;

f3 = 8e3;

Fs = 80e3;


% We need to find the period of the signal.

% From its DFT, we notice that it is the contribution of three cosine

% signals, and we know that the total period is the lcm of the three
```

```
% periods.
% Let's compute the periods in samples
P1_samples = Fs/f1;
P2_samples = Fs/f2;
P3_samples = Fs/f3;
P_samples = lcm(lcm(P1_samples, P2_samples), P3_samples);

% Signal definition:

% time axis in seconds
time = 0:1/Fs:10*P_samples/Fs - 1/Fs;

x = cos(2*pi*f1*time) + cos(2*pi*f2*time) + cos(2*pi*f3*time);

% Check the signal fft (not required)
X = fft(x);
N = length(x);
freq_axis = 0:Fs/N:Fs - Fs/N;
figure;
stem(freq_axis, abs(X), 'LineWidth', 2);
title('Absolute value of the DFT of the signal x(n)');
grid;
xlabel('f [Hz]');

%% 2. [4.5 pts]

% Design an FIR stop-band filter with the windowing method in order to
% remove the frequency f2 (but avoiding to remove f1).
% Consider 89 filter samples and, as cut-off frequencies, the +20% and -20%
% of the frequency to remove.
% Filter the signal x with this filter, defining the signal y.
% What do you expect to find in the DFT of the signal y? What will happen
% in the DFT behaviour if we increase the number of filter samples?
% Set the number of filter samples to 301 and repeat the filtering.
```

```
% n samples
N_filter = 89;
% filter order
filter_order = N_filter - 1;
% to find the cutoff frequencies, start from f2_norm
f2_norm = f2/Fs;
% 20% = 1/5
f_cutoff_1 = f2_norm - f2_norm/5;
f_cutoff_2 = f2_norm + f2_norm/5;

cutoffs = [f_cutoff_1, f_cutoff_2];
% matlab cutoff
cutoffs_matlab = cutoffs * 2;
h = fir1(filter_order, cutoffs_matlab, 'stop');
y = filter(h, 1, x);

% Filter behaviour (not required)
[H, omega] = freqz(h, 1, 1024, 'whole');
figure,
plot(omega./(2*pi)*Fs, abs(H));
title('|DTFT| of the filter H(z) (with 89 samples)');
grid;
xlabel('f [Hz]');

% Check the signal fft (not required)
Y = fft(y);
figure;
stem(freq_axis, abs(Y), 'LineWidth', 2);
title('Absolute value of the DFT of the signal y(n) (with 89 filter samples)');
grid;
xlabel('f [Hz]');

% What do you expect to find in the DFT of the signal y? What will happen
```

```
% in the DFT behaviour if we increase the number of filter samples?

% We expect to find that the contribution at f2 is attenuated. Since the
% filter is not a notch, the signal at f2 is not completely deleted, and
% the other contributions have their amplitude modified
% (especially f1 which is closer to f2).
% If we increase the number of filter samples, the filter will be more
% precise and it will allow to remove almost completely f2, while keeping
% almost untouched the other contributions.

% modify the number of samples to 301
N_filter = 301;
% filter order
filter_order = N_filter - 1;
h = fir1(filter_order, cutoffs_matlab, 'stop');
y = filter(h, 1, x);

% Filter behaviour (not required)
[H, omega] = freqz(h, 1, 1024, 'whole');
figure,
plot(omega./(2*pi)*Fs, abs(H));
title('|DTFT| of the filter H(z) (with 301 samples)');
grid;
xlabel('f [Hz]');

% Check the signal fft (not required)
Y = fft(y);
figure;
stem(freq_axis, abs(Y), 'LineWidth', 2);
title('Absolute value of the DFT of the signal y(n) (with 301 filter samples)');
grid;
xlabel('f [Hz]');

%% 3. [3 pt]
```

```
% We want to increase the sampling rate from 80KHz to 120KHz.
% Implement the sampling rate conversion of the signal y(n).
% You choose all the parameters needed.
% Define the new obtained signal as w.

% The ratio between 120 and 80 is 3/2.
L = 3;
M = 2;

% first, upsampling
y_upsampled = zeros(1, length(y) * L);
y_upsampled(1:L:end) = y;

% Define the low pass filter
cutoff = min([1/(2*L), 1/(2*M)]);
cutoff_matlab = 2*cutoff;
h = L*fir1(64, cutoff_matlab);

% filtering
y_f = filter(h, 1, y_upsampled);

% decimation
w = y_f(1:M:end);

%% 4. [2.5 pt]

% Plot the absolute value of the DFT of w as a function of the
% normalized frequency axis (keep the initial Fs as normalization frequency).
% Comment on the position of each peak you expect to find in W(f).
% Does the signal contain aliasing?

N = length(w);
W = fft(w);
```

```
freq_axis = 0:1/N:1 -1/N;


figure;

plot(freq_axis, abs(W));

grid;

title('|DFT| of the signal w(n)');

xlabel('f [norm]');


% The signal does not contain aliasing, since we introduced a filter in the

% middle of the process.

% We expect to find 4 peaks in one period of the spectrum.

% The position of the peaks will be related to +/- f1_norm * 2/3 and +/-

% f3_norm * 2/3.
```